What is claimed is:

1. A server for providing data to clients, the server comprising:

a dispatcher having a queue for storing requests received from clients; and

at least one back-end server;

wherein the dispatcher stores in the queue one or more of the requests received from clients when the back-end server is unavailable to process said one or more requests;

wherein the dispatcher retrieves said one or more requests from the queue for forwarding to the back-end server when the back-end server becomes available to process said one or more requests; and

wherein the dispatcher determines whether the back-end server is available to process said one or more requests by comparing a number of connections concurrently supported by the back-end server to a maximum number of concurrent connections that the back-end server is permitted to support, the maximum number being less than a maximum number of connections which the back-end server is capable of supporting concurrently.

2. The server of claim 2 wherein the dispatcher is configured to monitor a performance of the back-end server, to define the maximum number of concurrent connections that the back-end server is permitted to support, and to dynamically adjust the maximum number in response to the monitored performance.

3. The server of claim 1 wherein the server is a cluster-based server comprising a plurality of back-end servers, the dispatcher is configured to store in the queue said one or more requests when none of the back-end servers are available to process said one or more requests, and the

dispatcher is further configured to retrieve said one or more requests from the queue for forwarding to one of the back-end servers when said one of the back-end servers becomes available to process said one or more requests.

4. The server of claim 1 wherein the server is a Web server.

5. The server of claim 1 wherein the dispatcher and the back-end server are implementing using COTS hardware.

6. The server of claim 1 wherein the dispatcher comprises a first computer device, the back-end server comprises a second computer device, and the first and second computer devices are configured to communicate with one another over a computer network.

7. The server of claim 1 wherein the dispatcher is an OSI layer 7 dispatcher and said requests are data requests.

8. The server of claim 7 wherein the dispatcher implements a simplified TCP/IP protocol in user-space.

9. The server of claim 1 wherein the dispatcher is an OSI layer 4 dispatcher and said requests are connection requests.

10. A computer-readable medium having computer-executable instructions for performing the method of claim 1.

11. A method for controlled server loading, the method comprising the steps of:

defining a maximum number of concurrent connections that a server is permitted to support;

limiting a number of concurrent connections supported by the server to the maximum number;

monitoring the server's performance while it supports the concurrent connections; and

dynamically adjusting the maximum number as a function of the server's performance to thereby control a performance factor for the server.

12. The method of claim 11 wherein the defining step includes defining the maximum number to be less than a maximum number of connections which the server is capable of supporting concurrently.

13. The method of claim 11 wherein the concurrent connections are connections between the server and clients.

14. The method of claim 11 wherein the concurrent connections are connections between the server and a dispatcher.

15. The method of claim 11 wherein the server is a back-end server in a cluster-based server having a dispatcher, and the dynamically adjusting step includes dynamically adjusting the maximum number of concurrent connections that can be established between the back-end server and the dispatcher.

16. The method of claim 15 wherein each concurrent connection is a persistent connection over which data requests from multiple clients can be sent by the dispatcher to the back-end server.

17. The method of claim 11 wherein the dynamically adjusting step includes dynamically adjusting the maximum number in response to the monitoring step such that the server operates at or above a minimum performance level.

18. The method of claim 17 wherein the monitoring step includes monitoring the server's performance level in terms of a performance metric selected from the group consisting of request rate, average response time, maximum response time and server throughput.

19. A method for controlled server loading, the method comprising the steps of:

receiving a plurality of data requests from clients;

forwarding a number of the data requests to a server for processing; and

storing at least one of the data requests until the server completes processing at least one of the forwarded data requests.

20. The method of claim 19 further comprising the steps of retrieving the stored data request after the server completes processing at least one of the forwarded data requests, and forwarding the retrieved data request to the server for processing.

21. The method of claim 19 wherein the storing step includes storing a plurality of the data requests, the method further comprising the step of retrieving one of the stored data requests and forwarding the retrieved one of the data requests to the server for processing each time the server completes processing one of the forwarded data requests.

22. The method of claim 21 wherein the retrieving step includes retrieving the stored data requests on a FIFO basis.

23. The method of claim 19 wherein the data requests are HTTP requests.

24. The method of claim 19 wherein the receiving, forwarding and storing steps are performed by a single computer device having at least one processor.

25. The method of claim 24 wherein the single computer device comprises the server.

26. The method of claim 19 wherein the storing step is performed by a dispatcher and includes storing at least one of the data requests until the dispatcher receives a response from the server to at least one of the forwarded data requests.

27. A method for controlled server loading, the method comprising the steps of:

defining a maximum number of data requests that a server is permitted to process concurrently;

monitoring the server's performance; and

dynamically adjusting the maximum number in response to the monitoring step to thereby adjust the server's performance.

28. The method of claim 27 wherein the monitoring step includes monitoring the server's performance in terms of a performance metric selected from the group consisting of request rate, average response time, maximum response time, and server throughput.

29. The method of claim 27 further comprising the steps of receiving a plurality of data requests from clients, forwarding some of the data requests to the server for processing, and storing at least one of the data requests until the server completes processing one of the forwarded data requests.

30. The method of claim 27 wherein the defining step includes defining a maximum number of connections that can be supported concurrently by the server and limiting the number of data requests that can be pending on each connection.

31. The method of claim 30 wherein the defining step includes limiting the number of data requests that can be pending on each connection to one.

32. A method for controlled loading of a cluster-based server, the cluster-based server including a dispatcher and a plurality of back-end servers, the method comprising the steps of:

receiving at the dispatcher a plurality of data requests from clients;

forwarding a plurality of the data requests to each of the back-end servers for processing; and

storing at the dispatcher at least one of the data requests until one of the back-end servers completes processing one of the forwarded data requests.

33. The method of claim 32 wherein the storing step includes storing a plurality of the data requests and the forwarding step includes forwarding one of the stored data requests to one of the back-end servers each time one of the back-end servers completes processing one of the forwarded data requests.

34. The method of claim 32 wherein the cluster-based server is an L7/3 server.

35. A method for controlled loading of a cluster-based server, the cluster-based server including a dispatcher and a plurality of back-end servers, the method comprising the steps of:

defining, for each back-end server, a maximum number of data requests that can be processed concurrently;

monitoring the performance of each back-end server; and

dynamically adjusting the maximum number for at least one of the back-end servers in response to the monitoring step to thereby adjust the performance of the cluster-based server.

36. The method of claim 35 wherein the dynamically adjusting step includes dynamically adjusting the maximum number for each back-end server.

37. The method of claim 35 wherein the dynamically adjusting step includes dynamically adjusting the maximum number for said one of the back-end servers as a function of that back-end server's performance.

38. The method of claim 35 further comprising the steps of receiving a plurality of data requests from clients,

forwarding some of the data requests to the back-end servers for processing, and storing at least one of the data requests until one of the back-end servers completes processing one of the forwarded data requests.